

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-327042

(43)Date of publication of application : 12.12.1995

(51)Int.Cl.

H04L 12/44

(21)Application number : 07-062264

(71)Applicant : INTERNATL BUSINESS MACH
CORP <IBM>

(22)Date of filing : 22.03.1995

(72)Inventor : BERTIN OLIVIER
PRUVOST ALAIN
CHOBERT JEAN-PAUL

(30)Priority

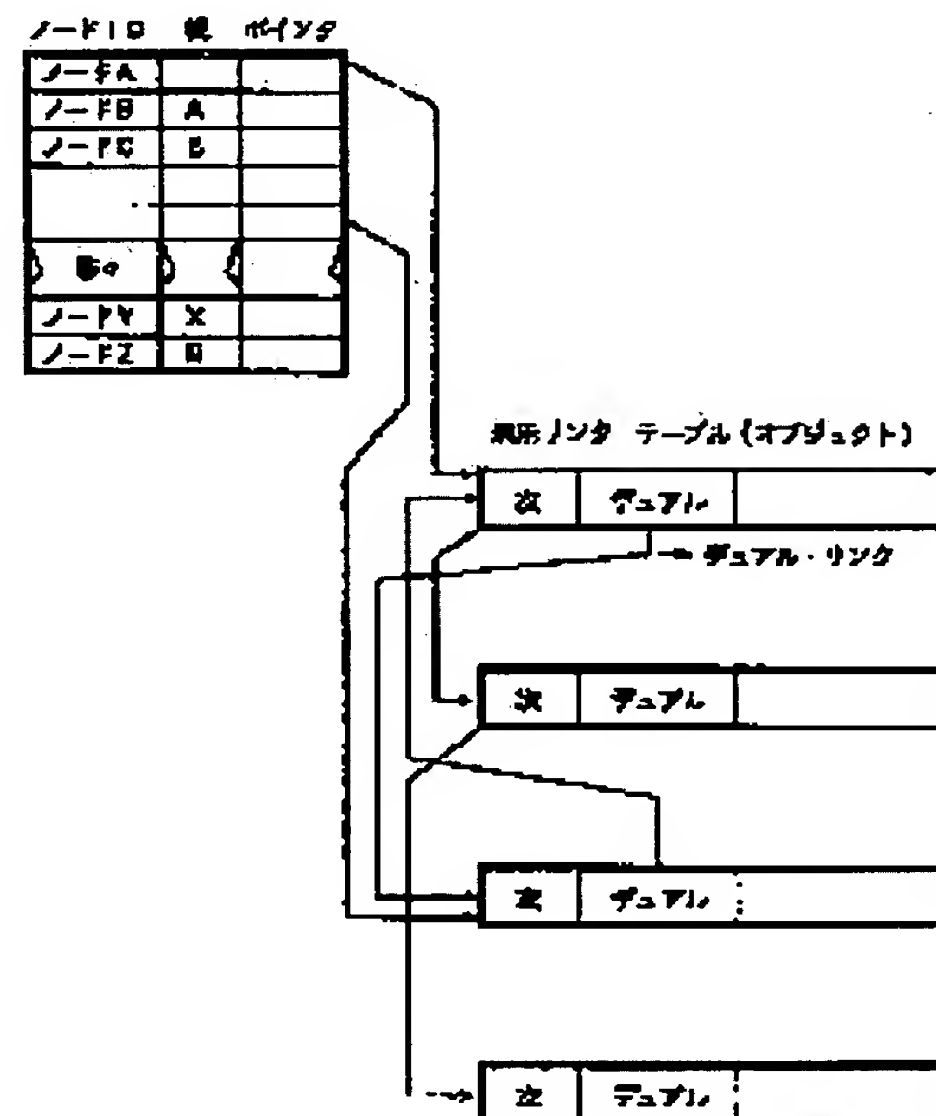
Priority number : 94 94480048 Priority date : 25.05.1994 Priority country : EP

(54) DATA COMMUNICATION NETWORK AND ITS NODE MANAGING METHOD

(57)Abstract:

PURPOSE: To attain optimum network application under various conditions including a rapid route determination and rapid spanning tree recovery.

CONSTITUTION: The system is provided with a means for allowing each node to dynamically set up and store a complete topology data base including a complete master node relation reference in its own node and provides spanning tree organization architecture in the data communication network interconnected by bidirectional links. The system is also provided with means for executing rapid route determination and rapid spanning tree recovery based on the contents of the topology data base.



LEGAL STATUS

[Date of request for examination]

25.04.1997

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

3149332

[Date of registration] 19.01.2001

[Number of appeal against examiner's decision
of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(51)Int.Cl.⁶
H04L 12/44

識別記号
F I

庁内整理番号
H04L 11/ 00

技術表示箇所
3 4 0

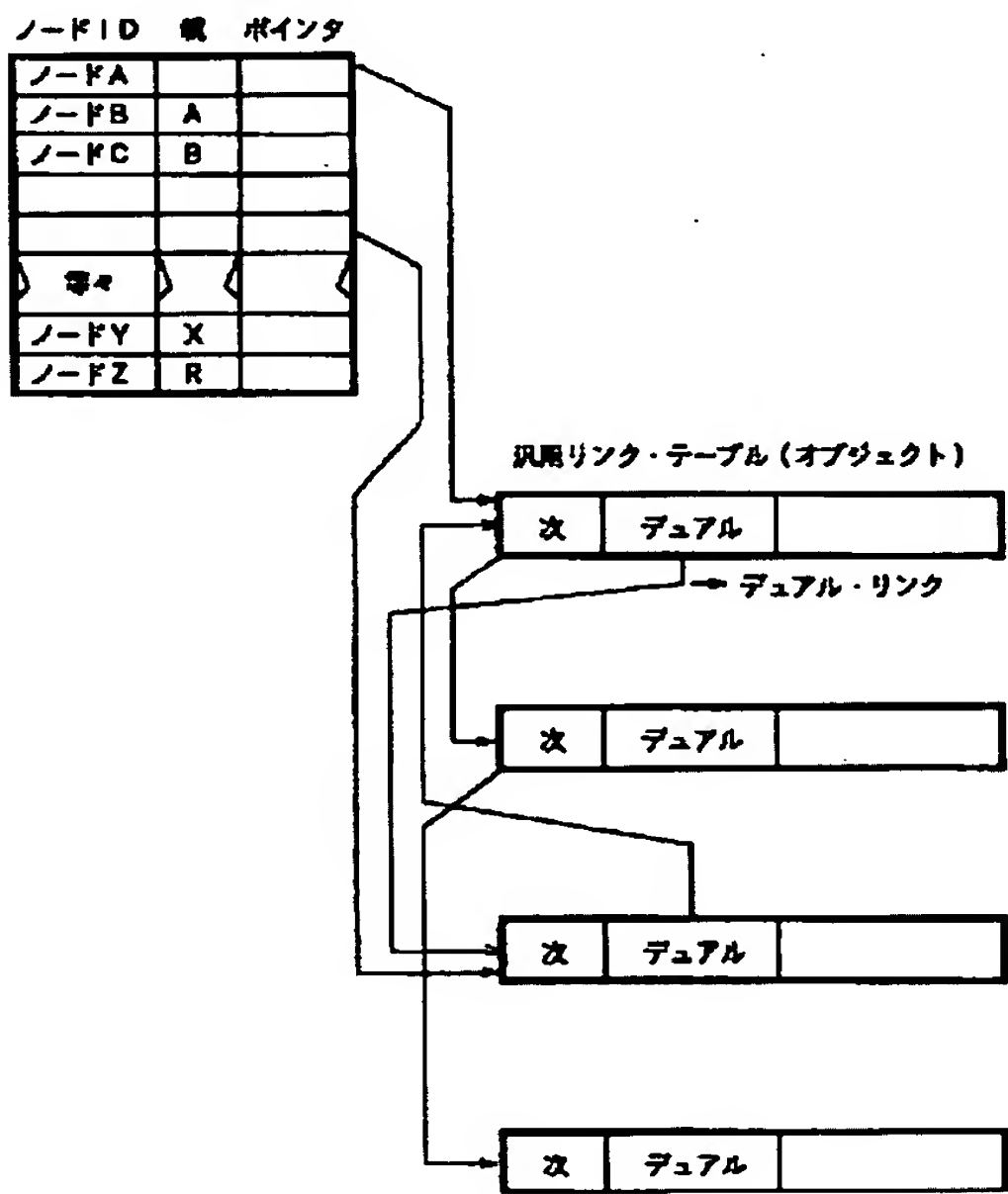
審査請求 未請求 請求項の数8 OL (全 13 頁)

(21)出願番号	特願平7-62264	(71)出願人	390009531 インターナショナル・ビジネス・マシーンズ・コーポレーション INTERNATIONAL BUSINESS MACHINES CORPORATION アメリカ合衆国10504、ニューヨーク州 アーモンク (番地なし)
(22)出願日	平成7年(1995)3月22日	(72)発明者	オリヴィエ・ベルタン フランス06200 ニース プールヴァール・ド・モンレアル 53
(31)優先権主張番号	94480048. 1	(74)代理人	弁理士 合田 潔 (外2名)
(32)優先日	1994年5月25日		
(33)優先権主張国	フランス(FR)		

最終頁に続く

(54)【発明の名称】 データ通信ネットワークおよびそのノードの管理方法

(57)【要約】
【目的】 迅速な経路決定および迅速なスパニング・ツリー回復を含む多くの状況において、最適な形でネットワークの運用を可能にする。
【構成】 ノードがそれぞれ、完全な親ノード関係の参照を含む完全なトポロジ・データベースをそのノード内部で動的に設定して記憶する手段を備え、双方向リンクによって相互接続された、データ通信ネットワークにおけるスパニング・ツリー編成アーキテクチャを提供する。このシステムはまた、前記トポロジ・データベースの内容に基づいて迅速な経路決定および迅速なスパニング・ツリー回復を行う手段を備える。



【特許請求の範囲】

【請求項1】双方向リンクによって相互接続された複数のノードを含み、任意の瞬間には、ルート・ノード以外の各ノードが単一の親ノードを有するスパニング・ツリー配列で相互接続された、ノード間でのデータ伝送のためのデータ通信ネットワークにおいて、前記スパニング・ツリーの各ノードが、

前記ツリー中のすべてのノードにつき、各ノードごとに、その親ノード情報とリンク・テーブルを指すためのアウトリンク・ポインタ手段とを含む親テーブルと、それぞれのデュアル・リンクを指すためのデュアル・リンク・ポインタ手段を含むリンク・テーブルと、前記リンクがスパニング・ツリーに現在関与しているかどうかを示すビット位置（STビット）を含む各リンクの参照を割り当てる手段とを含むトポロジ・データベースを記憶する手段と、

前記親テーブルとリンク・テーブルの内容を使って、ツリー配列を編成する手段と、

各スパニング・ツリーの編成時もしくは再編成時に、前記親テーブルと前記リンク・テーブルとを含むいわゆるトポロジ・データベースを動的に更新する手段とを含み、

前記各手段が通常の動作状態の間に動作可能である、データ通信ネットワーク。

【請求項2】各スパニング・ツリーの再編成時に前記親テーブルとリンク・テーブルを更新する前記手段が、

（a）当該ノードにおいて、ローカルに記憶された親テーブルとリンク・テーブルを走査して、子のない第1のノード即ち末端ノードを検出し、前記ノードをルートとして割り当てそれに応じてテーブルを修正する手段と、

（b）ローカルの親テーブルとリンク・テーブルとを含むトポロジ・データベースを調査して、ターゲット・ノードをその親とするスパニング・ツリー・リンクを見つけ、当該ターゲット・ノードをもつソース・ノードを親ノードとして選び、それに応じてテーブルを修正する手段と、（c）すべてのノードが親を獲得しそれに応じてテーブルを修正するまで、ソース・ノードをターゲットとしてその処理を繰り返し、当該ノードのデータベースにおける各ノードの完全な親子関係をツリー構造で定義する手段と、

を各ノードに含む請求項1に記載のデータ通信ネットワーク。

【請求項3】スパニング・ツリーの任意のノードがソース・ノードまたはターゲット・ノードのどちらかとして動作することができ、共にスパニング・ツリーに属するソース・ノードとターゲット・ノードとの間で、要求に応じて迅速な経路決定を行う手段をさらに含み、前記経路決定手段が、

ソース・ノードをトリガして、その記憶された親テーブルを上方に走査し、次々に親ノードを上方にリストする

ソース親リストを生成し記憶する手段と、

ソース・ノードをトリガして、その記憶された親テーブルを走査し、ターゲット・ノードから始まって上方にターゲット親リストを生成し記憶する手段と、

両方のリストを逆走査し、最後の共通ノードを除いたすべての共通ノードを削除する手段とを含み、

残りのソース親リストを順方向に、残りのターゲット親リストを逆方向に単純に連結することによって、前記ソース・ノードと前記ターゲット・ノードの間の経路を決定する、請求項1または2に記載のデータ通信ネットワーク。

【請求項4】スパニング・ツリーのリンク障害時に、元のスパニング・ツリーを、障害リンクの子ノードおよび当該子ノードに接続されているすべてのノードを含む第1のツリーと、元のルートおよび接続されている残りのすべてのノードを含む第2のツリーとに分割して区分し、回復操作を実行して、分割されたツリーを単一のツリーに再び結合し、対応するスパニング・ツリーを再定義する手段をさらに含み、前記回復手段が、

a) 障害リンクの前記子ノードを、前記第1のスパニング・ツリーのルートとして定義する手段と、

b) 元のルートと前記第1のツリーのルートの両方において、現ツリー区画を反映するようにそれぞれのデータベース内のSTビットを再設定する手段と、

c) 前記第1のツリーのルートにおいて、再設定されたトポロジ・データベースを走査して、前記第1のツリーのルートと前記第2のツリーに属するノードとの間のリンクをリストし、予め定義されたリンク特性に基づいて前記リンクのうちの1つを選択して、第1と第2のツリーを1つに接続し、それに応じて前記選択されたリンク上のターゲット・ノードに通知する手段と、

d) 前記第2のツリーのルート・ノード内で更新済みトポロジ・データベースを走査して、前記ターゲット・ノードと前記第1のツリーのルートとの間のリンクを選択する手段と、

e) ルート移動用の制御データを前記第2のツリーのルートからターゲット・ノードに送り、ツリー合併処理を完了させる手段と、

f) それに応じて各ノードのトポロジ・データベースを更新する手段と

を含む、請求項1ないし3のいずれか一項に記載のデータ通信ネットワーク。

【請求項5】障害リンクの子ノードをI、親ノードをJ、元のスパニング・ツリーのルート・ノードをRとして、ツリーのリンクIJの障害発生後に迅速なスパニング・ツリー回復動作を実施するために、Rをルートとする部分（R）として定義される区画と、Iをルートとする部分（I）として定義される区画の2つの区画に元のツリーを分割する、請求項1から4のいずれか一項に記載のネットワークにおいて実施される方法であって、

ノードIにおいて、
 Knを部分(R)に属するノードであるとして、トポロジ・データベースを走査してリンクIKnのリストを見つけ記憶するステップと、
 前記リストにおいて、予め定義されたリンク基準に基づいて、記憶されたリンクのうちの1つのリンクKを選択するステップと、
 前記ノードIにいわゆる結合要求メッセージを前記ノードKに送らせることによって、合併処理を開始するステップと、
 ノードRにおいて、
 Knを部分(R)に属するノードであるとして、トポロジ・データベースを走査してリンクKnIのリストを見つけ記憶するステップと、
 前記予め定義されたリンク基準に基づいて、1組のKnノードの中からリンクKIを選択するステップと、
 ルート移動メッセージをノードKに送るステップとノードKを使って合併処理を完了し、合併されたツリーのルート機能を獲得し、トポロジ・データベースの導入を再開するステップとを実行するノード管理方法。

【請求項6】前記合併処理開始ステップが、タイマを始動させて、ノードKに送られる結合要求メッセージの妥当性に制限を設けるステップを含む、請求項5に記載の方法。

【請求項7】前記ルート移動メッセージがノードRからノードKに送られる、請求項5または6に記載の方法。

【請求項8】多数のリンク障害が発生した場合に、リンク障害の発生順に従って操作を同期させる手段を使うことによってスパニング・ツリー回復動作を実施する、請求項5ないし7のいずれか一項に記載の方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、一般にデータ通信ネットワークに関し、より詳細には、迅速な経路決定および迅速なスパニング・ツリー回復を含む多くの状況において、最適な形でネットワークの運用を可能にする、いわゆるスパニング・ツリー編成のアーキテクチャ、およびそれを実施する方法に関する。さらに具体的には、本発明は、ツリー・トポロジを十分に反映するように各スパニング・ツリー・ノード・アーキテクチャを動的に編成する方法と手段、ならびに、前記編成ノード・アーキテクチャを使用して、たとえば迅速な経路決定や迅速なスパニング・ツリー回復などのネットワーク動作を実施する方法と手段に関する。この特定のノード編成は、従来のデータ通信ネットワーク動作中に生じる可能性があり、また実際に生じる他の多くの状況においても非常に有用である。

【0002】

【従来の技術】図1に、多数のノードA～Zを含むデータ・ネットワークを示す。

【0003】図に示したように、ネットワーク・ノードは、ノード間でデータ・パケットおよび制御データなど他の信号の伝送を可能にするために、リンクによって相互接続される。ネットワークの各リンクが、2つの隣り合ったノードを接続する。ネットワークの各リンクは、実際には、1対の単方向リンクまたはリンク・セグメントを備えることが好ましく、そのようなリンク対はそれぞれ、1本の光ファイバなどの単一通信媒体、または1対の同軸ケーブルや光ファイバなどの2つの別個の通信媒体によって実現できる。

【0004】そのような編成ネットワーク構造は、明らかに、いわゆるソース・ノードといわれるターゲット・ノードとを含む任意の1対のノード間でのデータの通信を可能にする。これらのソースとターゲットの指定はもちろントラフィックの方向によって動的に変化し、それが、図に示したすでに複雑なネットワーク・アーキテクチャとあいまって加わって、データおよび制御トラフィックの点からみたネットワーク動作が複雑になる。たとえば、データまたはトラフィック制御パケットが、ループで相互接続された1組のノードに偶然に経路指定されることがあり、そうすると、明らかにネットワークのスループットが低下し、あるいは一般にトラフィックが妨害されることになる。

【0005】場合によってはトラフィックの制御専用となっている、最適化されたネットワーク再編成はすでに当技術分野において開示されており、その再編成から、いわゆるスパニング・ツリー・アーキテクチャが導かれる。

【0006】図2には、通信ネットワークの任意のノード間でのメッセージ（たとえば、データまたは制御）の効率的な伝送を可能にするスパニング・ツリー、すなわち最適構造が示されている。スパニング・ツリーは、ネットワークが依然として接続されているがサイクルやループはまだないような、最小限の数のリンクを含む。したがって、データおよび制御パケットを同報通信するには適した媒体である。これらのリンクは、ルート・ノードと呼ばれる1つのノード（たとえば、図2ではノードA）から外側（終端側）に延びると考えることができる。スパニング・ツリーの外側ノード（たとえば、ノードF、ノードG、ノードK、ノードL、ノードN、ノードZ、ノードT、ノードW、ノードY）は、リーフ・ノードと呼ばれる。リーフ・ノードとルート・ノードの間のノードは、中間ノードと呼ばれる。図2のネットワークの構造においては、ノードA～Zのうちの任意のノードを、事例ごとに特定のツリー表現によるスパニング・ツリーのルート・ノードとみなすことが容易にできるとに留意されたい。

【0007】そのようなツリー構造においては、ネットワークの他のノードすなわち子ノードにルート・ノードとして知られるノードは、ネットワーク内に分散された

ハードウェアおよびソフトウェア手段を使いたいいわゆるルートシップ手順によって指定される。スパニング・ツリーの1ノード対を接続するリンクは「エッジ」と呼ばれることもある。

【0008】ルート・ノードは、それに直接接続されたノードの親ノードとも呼ばれ、ルート・ノードに直接接続されたすべてのノードは、そのルート・ノードの子ノードと呼ばれる。また、ツリー構造に沿って外に向かう連鎖において、別のノードの直接外側に接続されたノードはそれぞれ、前者のノードの子ノードと呼ばれ、前者のノードは後者のノードの親ノードと呼ばれる。すなわち、たとえば図2に示したスパニング・ツリー構造では、ノードAはノードB、H、Mの親であり、ノードBはノードCの親であり、ノードCはノードD、Gの親である。逆に、ノードD、Gは共にノードCの子ノードである。換言すると、所与のスパニング・ツリー構造において、ルート・ノードは、親ノードのない唯一のノードである。この特徴を別にすると、ルート・ノードは、スパニング・ツリーの他のどんなノードにも利用できない他のハードウェアまたはソフトウェア手段を持っている必要はほとんどない。したがって、ノードA～Zのうちの任意のノードが、所与の時間にルート・ノードとして動作することができる。

【0009】通常の動作条件下では、明らかに、スパニング・ツリーの再編成を必要とするいくつかの状況が生じることがある。たとえば、トラフィックを最適化するため、同様にたとえば、低速または小容量のリンクを高速または大容量のリンクで置き換えるため、あるいはノード間に新しい通信を設定するためのために、新しいリンクを設定する必要があることがある。そのようなリンクの置換えは、もちろん、トラフィック・ジャムを回避しまたはさらに悪い状況を回避するために、できるだけ速く実施できなければならない。

【0010】また、あるノードに接続された端末が別のノードに接続されたCPU（中央演算処理装置）へのアクセスを要求すると仮定すると、システムは、スパニング・ツリーのエッジを使って、転送ネットワークのこれら2つのノード間の簡単で速い経路選択手順を提供しなければならない。この場合にも、経路が速く設定されるほど、ネットワークの動作がよくなる。

【0011】さらに、現代のデータ・ネットワークにおけるトラフィックの重要性を考えると、どんな障害（たとえば、リンク障害）が生じて、スパニング・ツリーの一部が分離されて、そこにあるトラフィックが完全に麻痺し、すなわちデータ・ネットワーク全体の動作が麻痺することになる。これらの状況は、従来技術においてすでに考慮されているが、提案された解決策はスパニング・ツリーを再編成するために必要な時間の点で十分満足できるものとは考えられない。そのようなリンク障害の結果がいかに劇的なものであるかは、最初のリンクが

設定される前に新しいリンクの障害が発生する場合を考えてみればはっきりとわかる。その場合、トラフィックは完全に麻痺し、損失（たとえば金銭の損失）は莫大なものになるであろう。

【0012】マルチメディア環境におけるこの種の状況は、特にデータ量の大きさ、データの複雑さゆえに特に問題となる。

【0013】したがって、上記その他の問題に対する良い解決策があれば、特に高価でない手段で実施できるのであれば、データ・トラフィック業界から、さらにはユーザからも最も歓迎されることになる。これがまさに本発明が意図することである。

【0014】

【発明が解決しようとする課題】本発明の1つの目的は、データ通信ネットワーク・アーキテクチャ、より詳細には、ほとんどの動作条件下で迅速なスパニング・ツリー、ノード動作またはツリー再編成を可能にする派生スパニング・ツリー・アーキテクチャを提供することである。

【0015】本発明の他の目的は、任意のツリー修正後のノードの自己設定を可能にするノード編成とそれに対応する手段を提供することである。

【0016】本発明の他の目的は、スパニング・ツリーの2つのノード間で要求された経路を迅速に設定するための方法および手段を提供することである。

【0017】本発明の他の目的は、リンク障害のケースを含む従来の多くの動作条件下で、スパニング・ツリーを、しかもかなり迅速に再編成するための方法および手段を提供することである。

【0018】本発明の他の目的は、リンク障害が多数ある場合でも、スパニング・ツリーの迅速な回復を可能にする方法とそれに対応する手段を提供することである。

【0019】

【課題を解決するための手段】これらの目的は、双方向リンクによって相互接続された複数のノードを含み、任意の所与の瞬間に、親子関係で外側に向かって編成された（ルート・ノード以外の）各ノードが単一の親ノードを有するようなルート・ノードと子ノードとを含むスパニング・ツリー配列として相互接続され、ノード間でのデータ・パケットおよび制御データの伝送を可能にするデータ通信ネットワークにより達成される。このスパニング・ツリーにおける（ルート・ノード以外の）各ノードは、その親ノード情報とリンク・テーブルを指示するためのアウトリンク・ポインタ手段とを含む親テーブルと、それぞれのデュアル・リンクを指示するためのデュアル・リンク・ポインタ手段とを含むリンク・テーブルと、リンクがスパニング・ツリーに現在関与しているかどうかを示すビット位置（STビット）を含む各リンク参照を割り当てる手段とを含むトポロジ・データベースを記憶する記憶手段を具備するとともに、親テーブルと

リンク・テーブルの内容を使って、ツリー配列を自由自在に編成する手段と、各スパニング・ツリーの(再)編成時に、親テーブルとリンク・テーブルとを含むいわゆるトポロジ・データベースを動的に更新する手段とを含んでいる。

【0020】

【実施例】以下の説明は、図1に示したような一般のデータ・ネットワークに基づいて行う。このネットワーク中では、所与の瞬間に、いくつかのリンクが動作しており、他のリンクは動作していない。対応するスパニング・ツリーは、所与の瞬間にノードA～Zを接続するエッジを有する、図2に示したものである。

【0021】すでに述べたことであり、データ通信ネットワークの当業者には明らかであるが、図1と図2のネットワークの図は共に本発明を例示するために示したものにすぎず、本発明の範囲を限定するものと考えべきではない。ネットワークは、実際にはもっと複雑なものでももっと簡単なものでもよい。

【0022】また、ノード間の距離は、比較的近距离から比較的遠距離(たとえば、数千キロメートル)まで変化してもよい。

【0023】また、従来のデータ端末とメインフレームまたはCPUとを含む端末(図示せず)が所与のノードに接続されて、従来のログオン手順を使ってデータ・トラフィックを要求し制御することを理解されたい。これらの手順は、本発明に直接関係しないので、本明細書では説明しない。

【0024】最後に、ネットワークのノードおよびリンクは、電力や速度、トラフィック容量などの点で必ずしも互いに等価ではないが、そのことが、本発明を妨げたり影響を与えたりすることはない。

【0025】また、親、子およびルートの関係は、前述のように定義され、スパニング・ツリーの表現は、時間と共に変化することができ、おそらくは実際に変化し、ノードのどれかがスパニング・ツリーのルートになる。

【0026】すでに述べたように、データおよび制御情報はネットワークのいたるところに経路指定される。これらの情報は、一般に純粋なデータと制御データとを含めてデータと呼ばれ、従来のパケット構造に個別に編成される。各パケットは、経路指定情報(アドレス)を含む、いわゆるヘッダ・セクションを有する。この経路指定情報は、ネットワークのリンクとノードを介してネットワークのいたるところに当該パケットをソース・ポイントから指定されたターゲット・ポイントまで運ぶためにネットワークによって使用される。

【0027】複数のノードに同報通信またはマルチキャストされるパケットもある。換言すると、マルチキャスト経路指定ノードは、1つのノードから複数の受信ノードに同時にパケットを送ることを可能にする。送信ノードおよびそれに対応する受信ノードは、マルチキャスト

・ツリーとして動作する。マルチキャスト・ツリーが定義されると、ツリー・アドレスが、そのツリーに関連付けられると考えることができる。様々なネットワーク・ノードは、リンク・ハードウェアにおいて適切なツリー・アドレスを設定することにより、その関連リンク(リンクの所有権が定義された)のうちのどれがマルチキャスト・ツリー上にあるかを指定する。

【0028】ネットワーク中で伝送されたパケットは、その経路指定ヘッダ内にツリー・アドレスを含む。これらのパケットは、ネットワーク中で、そのハードウェア内に正しいツリー・アドレス・セットを有するマルチキャスト・ツリーのリンク部分上だけに伝播される。

【0029】マルチキャスト・ツリー編成とマルチキャスト経路指定の例は、図3と図4の両方に示されている。

【0030】図3には、それぞれノード1、2、3、4と名付けられた4つのノードを含むネットワークが示されている。これらのノードは、物理リンクによって相互接続されており、各リンクは2つの単方向リンクからなる。それぞれの単方向リンクは、その起点ノードによって定義される(「属する」とも言える)。逆に、そのノードはリンクを「所有する」と言われる。

【0031】すなわち、ノード1は、2つのリンク、リンク1-aと1-bを所有し、ノード2は、3つのリンク、リンク2-a、2-b、2-cを所有し、ノード3は、2つのリンク、リンク3-aと3-bを所有し、ノード4は、3つのリンク、リンク4-a、4-b、4-cを所有する。

【0032】図3に示したように、リンク4-aは、リンク1-bのデュアル・リンクであると言われ、両方のリンクがノード1とノード4の間で完全な物理リンクを形成する。

【0033】マルチキャスト・ツリーは、先に検討した、点線のリンクによって相互接続された4つのノードで編成される(これは可能な表現のほんの一例にすぎない)。前記のマルチキャスト・ツリーは、前もって割り当てられたツリー・アドレス"TA"によって定義される。ノード1が、その経路指定ヘッダ・セクション内にアドレスTAをもつパケットを送るとき、前記パケットは自動的にリンク1-aおよびリンク1-b上に送られ、次にリンク4-cを経てノード4からノード3に経路指定される。マルチキャスト・ツリーの一部分であるリンクだけが、当該のパケットを伝播する。最後に、ノード1からのパケットは、要求に応じてマルチキャスト・ツリーを介してノード2、3、4に実際に同報通信される。

【0034】図4は、ノード4においてマルチキャスト・ツリー"TA"を支援する交換ハードウェアの概略図である。ノード4は、ノード4に接続された各リンクに対応する多数のポート、すなわちポート4-a、ポート4

ーb、ポート4ーcを備える。各ポートは、マルチキャスト・ツリー・アドレスを記憶するバッファを備え、バケットは、入力ポートとして動作するポート以外の、対応するツリー・アドレスをもつヘッダを有する当該バケットを支援するポートを介してのみマルチキャストされる。

【0035】換言すると、経路指定ヘッダ内に"TA"をもつバケットがノード4で受け取られるとき、そのノードの内部交換ハードウェアは、バケットがそのノードに到達するのに使用されたリンク以外の、"TA"マルチキャスト・ツリーの一部としてマークされたすべてのリンクに、前記バケットのコピーを与える。今の場合、バケットはリンク4ーcに転送される。

【0036】スパニング・ツリー(ST)は、ある種の動作条件下では、ネットワーク内のすべてのノードを結合するマルチキャスト・ツリーの1つの例として動作する(図1と図2を参照)。ノードによって送られ、経路指定ヘッダ・セクション内にスパニング・ツリー・アドレスをもつバケットは、上記のマルチキャスト処理およびハードウェアを使用することによって、ネットワークの他のすべてのノードに自動的に送られる。

【0037】この機構は、すべてのネットワーク・ノードに迅速に到達しなければならないネットワーク制御メッセージを、元のデータ通信ネットワークのすべてのリンク上で同報通信することなしに伝播するのに特に有用である。同報通信の場合は、メッセージが各ノードによって複数回受け取られることになる。したがって、スパニング・ツリーにより、スパニング・ツリーの送信ノード部分によって送られるメッセージが、スパニング・ツリーに属する各受信ノードによって1回だけ受け取れることが保証される。

【0038】上記の考察によれば、スパニング・ツリー動作は、多少とも迅速に、より一般的に言えば、提供された追加のノード・アーキテクチャに応じて多少とも最適化された形で動作することになる。

【0039】本発明において、各ノードは、完全なスパニング・ツリー表現またはトポロジ・データベースを設定する手段と、この完全なトポロジ表現を極めて迅速に動的に再調整する手段とを備える。そのために、各ノードは、従来の既存のプロセッサ手段およびランダム・アクセス・メモリ手段と、既に述べた図3と図4のハードウェアおよび論理とを使用して、前記スパニング・ツリーの完全な表現を有するノードを編成して維持し、ネットワークの最適動作を可能にするソフトウェア手段を備える。

【0040】より正確には、各ノードは、制御装置と、トポロジ・データベースを記憶するランダム・アクセス・メモリ手段とを含む制御点アダプタ論理機構を備える。トポロジ・データベースは、ネットワーク・ノードのリストを作成する。その際に各ノードが実際のスパニ

ング・ツリー表現の親ノードを指定し、アウトリンク・ポインタがアウトリンク・テーブルを指す。このアウトリンク・テーブルは、デュアル・リンク・テーブルを指すデュアル・リンク・ポインタと、当該リンクが実際にスパニング・ツリー(ST)上にあるかどうかを示すビット位置(STビット)とを含む。

【0041】たとえば、ノードCを検討すると仮定する。このノードは、それぞれ他のノードと同様に、その中に記憶されたいわゆる親テーブル(ノード・テーブルまたはリスト)を含む。前記テーブルは、ノード・リスト形式で、ノードA、ノードB、ノードC、ノードD、…、ノードZ(すなわち、すべてネットワーク・ノード)をリストする。各ノードは親ノードを指す。各ノードはまた、第1のリスト即ちアウトリンク・テーブルを指す。たとえば、ノードCは、CB、CD、CG、CIを含むリストを指す(図1参照:これらのリンクは、CBがCDを指し、CDがCGを指し、CGがCIを指すことによってともに連結される)。

【0042】また、前記第1のテーブルには、当該リンクが実際のスパニング・ツリーに属するかどうかに関する指示が記述されている。前述の通り、単一ビット(ST)がそのジョブを行う。前記STビットが2進値"1"に設定されている場合は、前記リンクがスパニング・ツリーに属することを示す。逆に、STが"0"にリセットされている場合は、リンクはツリーから離れている。実際には、図2の説明では、リンクCB、CD、CGについてはST=1、リンクCIについてはST=0である。

【0043】このとき、第1のリストの各リンクが、第2のリスト即ちデュアル・リンク・テーブルを指す。たとえば、CBは、BC、BI、BA、BDを含むリストを指す。デュアルな状況であるため、第2のテーブルのBCは、第1のリストのCBを逆に指すポインタを含むことは明らかである。前記第2のリストはまた、各リンクがスパニング・ツリー上にあるかどうかを示す。したがって、STビットは、BCとBAの前で"1"に設定され、一方BIとBDの前では2進値ゼロである。

【0044】次に、前記第2のリスト内の各リンクは、第3のリスト即ちデュアル・リストを指す。たとえば、BAは、AD、AB、AI、AH、AM、AQを含む第3のリストを指す。前記第3のリストにおける次のリンク、すなわちAB、AH、AMは、2進数の値1に設定されたSTビットを有し、一方、残りのリンクはST=0である。

【0045】以下同様に、ネットワークが完全に記述されるまで同じ操作が続く。(実際には、これらのリンク・テーブルまたはリストをすべて組み合わせて単一のリンク・テーブルにすることもできる)。実際には前記表現は、当該ノードから見るとわかるようにノードに依存しているが、各ノードはあらゆるノードの親子関係を含

む完全なツリー表現を含んでいる。これらのテーブルは、実際には、組み合わせて図5に概略的に示したトポロジ・データベースにすることができる。

【0046】図6～9には、ローカルのスパニング・ツリー表現を構築するためのフローチャートが示されている。図6は、スパニング・ツリーのローカル・イメージのルートを決する方法を示す。ローカル・ルートは、次々に親を選択するときに出会う最初の末端ノード（子ノードのないノード）である。

【0047】所与のノードX'にいと仮定する。処理は、段階50で始まり、所与のノードX'がそれ自体ノード・リストに記録される。次に、段階51で、対応する親ノードY'がノード・リストに入れられる。次に、システムはノードY'の最初のリンクを獲得し（段階52）、前記リンク上でテストを開始して（段階53）、それがスパニング・ツリー上にあり、オンラインであり、X'から来ていないかどうかを検査する。段階53のテストの結果が肯定の場合は、宛先ノードがY'の親に設定され、ノード・リストに記録される。X'はY'と同じに設定され、新しいY'がY'の親になる（段階54）。次に、処理は段階52に戻る。

【0048】逆に段階53のテストが否定であった場合は、第2のテストを行って（段階55）、当該リンクが前記ノードの最後のリンクであったかどうかを判定する。段階55で行われたテストの答えが否定の場合、システムはノードY'の次のリンクを検討する（段階56）。そうでない場合は、段階57に進み、前記ノードY'がローカル・ルートとして記録される。

【0049】次に、処理は、ノード・テーブルとデータベースを走査して対応する親情報を更新するためのアルゴリズムを表す図7～9に進む。最初に、ルートがノード・リストから取り除かれ（段階61）、テストを行ってノード・リストが空かどうかを検査する（段階62）。そうである場合は、ローカル・スパニング・ツリーは完全に記述され、親ノードを有する、ローカルのルート以外のすべてのノードが定義され記録される。そうでない場合は、ノード・テーブルの第1のノード（たとえば、ノードX'）を読み取り（段階63）、テストを行って、前記ノードX'が親ノードを獲得したかどうかを検査する（段階64）。このテストの答えが否定の場合は、第1のリンクを獲得し（段階65）、前記リンクがスパニング・ツリー（ST=1）上にありオンラインであるかどうかをテストする（段階66）。段階66のテストの答えが否定の場合は、テストを行って（段階67）、当該リンクが検討すべき最後のリンクであったかどうかを検査する。そうでない場合は、次のリンクが検討され、段階66に戻る。段階66のテストの答えが肯定の場合は、システムは、宛先ノードが親を有するかどうかをテストし（段階68）、親がない場合は、テスト段階67に進む。そうでない場合は、宛先ノード（たと

えばY')がノードX'の親として設定され、Y'がノード・リストに記録される（段階69）。テスト段階64と67のどちらかの結果が肯定の場合、および段階69の動作の完了時に、別のテストを行って（段階70）、当該ノードがノード・テーブル内で走査すべき最後のノードであるかどうかを検査する。

【0050】この最後のテストの答えが否定の場合、システムは、ノード・テーブルの走査を続けて、その次に記録されたノードを検討し（段階71）、段階64に戻る。答えが肯定の場合、システムはすべてのノードが親を獲得したかどうかを検査し（段階72）、獲得していなければ、段階62に戻り、獲得していれば、ノード・リストを除去して同じ段階62に戻る。

【0051】図6～9のフローチャートから、プログラミングの分野の当業者なら、システム（たとえば、プロセッサ）の特性が与えられれば、発明的な努力を必要とせずに、適切なプログラミング言語によるプログラムを簡単に誘導できることは明らかである。

【0052】すでに述べたように、本発明によって提供される完全ローカル・ツリー表現は、いくつかのクリティカルな状況において、対応するあらゆる問題の解決を大幅に簡素化しかつ高速化することにより、ネットワークおよびスパニング・ツリーを改善する。

【0053】たとえば、簡単に述べると、スパニング・ツリーのエッジ（リンク）のみを使って、いわゆるソース・ノードといわゆるターゲット・ノードとの間で経路を迅速に決定する事例を検討しよう。迅速な経路決定はシステムによって開始される。この場合、この動作は、ソース・ノードのトポロジ・データベースをソース・ノードとターゲット・ノードの両方についてルート・ノードまで調査し、ルート・ノードまでのすべてのターゲットの親ノードをリストするとともに、ルート・ノードまですべてのソースの親ノードをリストすることによって実行される。2つのリストの最初の共通ノードが、経路を定義するために使用される。

【0054】たとえば、ソース・ノードZが、ターゲット・ノードXに向かうスパニング・ツリーの経路を確立しようとしているものと仮定する。

【0055】ノードZ内に設定された論理は、ローカル・テーブルを走査して、以下のリストを確立する（図2参照）。

ソース-親リスト=Z, R, Q, P, O, M, A
ターゲット-親リスト=X, U, R, Q, P, O, M, A

【0056】次に、システムは、上記両リストの逆走査を続けて、最後の共通ノードを除くすべての共通ノードを削除する。その結果、両方のリストで、Aが最初に削除され、次にM、O、PそしてQが削除される。最後の共通ノードはRであり、ノードZは、残りのソース・リストを順方向にまたターゲット・リストを逆方向に局所

的に走査することにより、探していた経路をZ、R、U、Xとして設定し、制御メッセージ中でそれを定義する。

【0057】しかし、本発明は、リンクがたまたま切断し、スパニング・ツリーが極めて迅速に回復する（再定義される）必要のある状況でよりいっそう重要である。この状況がいかに劇的であるかを理解するには、ツリー内で同時に複数のエッジが破裂した場合、さらには、前記破裂が生じた後ツリーが再定義される前に次々に破裂が十分迅速に起こることを考えてみるだけでよい。強調するまでもないが、そのような状況から派生する混乱により、ネットワークがほとんど完全にダウンし正常に動

かなくなる。

【0058】本発明によって提供される前記トポロジ・データベースを各ノードに含ませる完全なツリー表現により、本システムは既知のほとんどのシステムよりも迅速に回復する。

【0059】以下に、迅速なスパニング・ツリー回復を操作する本発明を例示する。問題を十分に理解するには、先に開示した分散ツリー保守アルゴリズムが、ルートと呼ばれる唯一モードのノードを定義することを理解しなくてはならない。ルートは、スパニング・ツリーのほとんどの保守を集中し調整する。

【0060】エッジ障害が発生したとき、元のスパニング・ツリーは2つのツリーの区画に分割される。障害の前に子だったノードが、新しく作成されたツリー区画のルート（親のないノード）になる。各ノード内に以前から設けられている手段、すなわち完全なツリー記述によって、特に障害の起こったエッジもわかっているとき、各区画はスパニング・ツリーの区画の回復を試みることができる。しかし障害から回復するには、最終的には2つの区画を再び1つのツリーに合併しなければならない。しかし、2つに分割されたツリーを合併するには、それぞれのルートが、同じエッジがそれらを相互接続するのに適切であると合意しなければならない。

【0061】ネットワーク・トポロジが変化した（リンクとノードの障害および回復）後にスパニング・ツリーを更新する方法は、イズリアル・シドン（Israel Cidon）、インダー・S・ゴペール（Inder S. Gopeel）、シャイ・クッテン（Shay Kutten）、およびマルシア・ペータ（Marcia Peter）の論文"Distributed Tree Maintenance", IBMテクニカル・ディスクロージャ・ブルテン、Vol.35, No.1A, 1992年6月、pp.93~398に記載されている。その開示では、トポロジ保守アルゴリズムは、ツリー上のノードがその隣接エッジのトポロジを結局は知ること保証する。相互接続のためのルート・ノード間の合意は、それらを厳密に順序づけできるようにすべてのネットワークのエッジに一義的に割り当てられた「重み」と呼ばれる特性によって保証される。2つのツリーを合併するには、各ツリーのルートが、重みが最も

小さいそれらの間の非ツリー・エッジを合意しなければならない。次に、ルートを、Move-Rootと呼ばれる信頼できる2地点間メッセージによって、あるノードから別のノードに移動する。Move-Rootを送ることができるノードだけが、現ノードである。Move-Rootメッセージを送ると、現ルートはもはやルートではない。Move-Rootの受取側が、次のルートになる。トポロジ更新アルゴリズムとツリー保守アルゴリズムは、現ルートがネットワーク内のすべてのエッジを検討してルートをどこに移動するかを決定するという点で相互依存的である。

【0062】ツリー・トポロジの十分な知識、より詳細には本発明によって提供される各ノードにおけるノードの親子関係の知識が、前述したような、例から導かれる迅速な経路決定とあいまって、より迅速なツリー回復を可能にする。システムによって開始される、この迅速なスパニング・ツリー回復と名付けられた処理を以下に説明し1つの例を示す。

【0063】エッジBCに障害が起こったと仮定する。元のツリー（図2参照）が2つのツリーに区分される。元のツリー（ツリー（A））はノードC、G、D、E、Fを除外し、新しいツリーはこれらのノードC、G、D、E、Fだけを含む。定義により、障害が起こったエッジの子ノードつまりノードCが、ルートに設定される（ルート自己設定）と仮定すると、定義により、ノードC、D、E、F、Gを含むツリーに関して、この第2のツリーをツリー（C）とラベルすることができる。

【0064】2つの区画、すなわちツリー（A）とツリー（C）を合併する処理は、ノードCがそのトポロジ・データベースを走査することから始まる。

【0065】前記データベースを下記に概略的に示すが、これは、実際に容易に組み合わせて単一のリンク・テーブルにすることのできる、親テーブルとリンク・テーブルを示す。

【0066】ノードCにおける親テーブルは、ネットワーク・ノードを、各ノードの前に対応する親ノードをつけてリストする。したがって、リンク障害の前には、ルートであるノードAは親がない。ノードBの親はAであり、ノードCの親はBであり、以下同様にしてZに達する。したがって、リンク障害前の親テーブルは以下のようになる。

【0067】

【表1】

15
親テーブル

ノード	親	アウトリンク・ポインタ
ノードA	なし	:
ノードB	A	:
ノードC	B	---->リンク・テーブルへ
:	:	:
ノードX	U	:
ノードY	X	:
ノードZ	R	:

* ウトリンク・ポインタをも含み、前記ポインタはそれぞれのノードから出るリンクを指し（第1のリスト）、各リンク・リストはデュアル・リンク・リスト（第2のリスト、第3のリストなど）を指す。

【0069】次に、ノードCからのアウトリンクを示す第1のリストと、CBデュアル・ポインタ（PTR）が指すデュアル・リストを示す第2のリストと、BAデュアル・ポインタ（PTR）が指す第3のリストとを有する1つの例を検討する。

10 【0070】

【0068】図のように、親テーブルはノードごとにア＊

【表2】

リンク・テーブル

第1リスト	リンク	リンク・ポインタ	デュアルPTR	STビット
:	:	:	:	:
:	CB	:	:	1
:	CD	:	:	1
:	CG	:	:	1
:	CI	:	:	0

【0071】

※20※【表3】

第2リスト				STビット
:	:	:	:	:
:	BC	:	:	1
:	BI	:	:	0
:	BA	:	:	1
:	BD	:	:	0
:	:	:	:	:

【0072】

★ ★【表4】

第3リスト				STビット
:	:	:	:	:
:	AD	:	:	0
:	AB	:	:	1
:	AI	:	:	0
:	AH	:	:	1
:	AM	:	:	1
:	AQ	:	:	0

【0073】上記のように、ノードC参照の前にあるアウトリンク・ポインタは、それぞれ次のリンク（上のテーブルには図示せず）を指すCB、CD、CG、CIを含むリストである第1のリンク・リストを指す。換言すれば、CBは、CDを指し、CDはCGを指し、以下同様にしてCIに達する。さらに、すでに述べたように、第1のリンク・リストは、第2のリンク・リストを指すデュアル・リンク・ポインタを含む。たとえば、第1の

リンク・リスト中のCBに関連するデュアル・ポインタは、リンクBC、BI、BA、BDをリストしたデュアル・リスト（または第2のリンク・リスト）を指す。前記第2のリンク・リストにおける各リンクの参照は、リンクBAのデュアル・ポインタに関して先に示したのと同様に、第3のリンク・リストを指し、以下同様である。

50 【0074】これらすべてのリンク・リストは、実際に

は、単一の複雑なリンク・テーブルにまとめられる。

【0075】さらに、前述のリンク障害回復の助けとなるはずであるが、各リンクは、対応するリンクが現在スパニング・ツリー上にあるか（ST=1）それともスパニング・ツリー上にないか（ST=0）を示す1ビットの参照（すなわちST）によって定義される。たとえば、図2に示した元のツリーに関して、かつノードCに関して、第1のリンク・リストは、CB、CD、CGがST=1、CIがST=0であることを示す。

【0076】リンクBC/CBに障害が起こると、上記のように、ノードC、すなわち障害エッジ上の子ノードが、C、G、D、E、Fを含むツリーのルート・ノードになる。ツリー（C）上のこれらのリンク、すなわちCG、CD、DE、EFだけでSTビットが2進値1に保たれ、他のリンクはリセットされる。したがって、上記第1のテーブルは以下になる。

【0077】

【表5】

リンク:	: ST
CB :	: 0
CD :	: 1
CG :	: 1
CI :	: 0

【0078】また、親テーブルは、リンク障害とツリーの区分を反映するようにリセットされる。

【0079】上記の情報が与えられているものとする、ノードCはそのトポロジ・データベースを単に走査し、特にリンク・テーブルを走査することによって、リンクCKnのリストを決定することになる（ただしKnは、ツリー（A）に属するノードである）。上の例では、リンクCKnのリストは、ツリー（C）に属さないリンクとしてCBとCIだけを含み、したがって、ツリー（A）とツリー（C）の両方を相互接続するために使用できる可能性がある。

【0080】CBに障害が起こったため、残りのリンクのリストはCIだけを含み、そのCIが選択のために指定される。より一般的に言えば、CKnのリストはいくつかのリンクを含む。この場合、あいまいさを高めリンクを選択するために追加のパラメータが事前定義することもできる（たとえば、最も高速のリンク）。

【0081】したがって、ノードCによって選択されるリンクはCIである。ノードCは、いわゆる「結合要求」メッセージを送ることによって、ノードIとの対話を開始する。一方、ノードIは、ツリー（A）のルートと認められた後でのみ応答する。この場合、ツリー（A）はツリー（I）になる。ノードIは、ルートとなった時にのみ結合要求を受け入れ、ここではまだそうっていない。問題が生じた場合に無期限に待つことを選

けるために、ノードCはタイマをセットし、前記設定時間だけ待機する。

【0082】また、ツリー（A）上で同様の処理（テーブルのリセットなど）を用いることにより、区分された2つのツリーの間で設定すべきリンクが選択できるようになる。換言すると、リンクBCに障害が起こったというメッセージを受け取った後、元のルート（A）は、そのトポロジ・データベースを走査して、候補リンクKnCのリストを見つける。このKnは、ツリーの分裂が起こった後は、ツリー（A）、すなわちC、G、D、E、Fを除くネットワークのすべてのノードを含むツリーに属するノードである。

【0083】1組のCKn中で同じリンクICが、ルートAによって選択される。

【0084】したがって、前述のような本発明の迅速経路決定手順を使用することにより（親リストを走査するだけで）、ノードAからノードIへの即時のMove-Rootが実施できる。ノードAは、ノードAとノードIの間のスパニング・ツリー上で選択された経路を使ってノードIにMove-Rootメッセージを発行する。次に、ノードIは、ノードCからCIを介して受け取った「結合要求」メッセージを受け入れ、Cとの合併処理を完了する。その後、ノードIを新しいツリーのルートとし、上記の処理に従って合一化されたスパニング・ツリーが再定義される。

【0085】まとめとして、本発明の構成に関して以下の事項を開示する。

【0086】（1）双方向リンクによって相互接続された複数のノードを含み、任意の瞬間には、ルート・ノード以外の各ノードが単一の親ノードを有するスパニング・ツリー配列で相互接続された、ノード間でのデータ伝送のためのデータ通信ネットワークにおいて、前記スパニング・ツリーの各ノードが、前記ツリー中のすべてのノードにつき、各ノードごとに、その親ノード情報とリンク・テーブルを指すためのアウトリンク・ポインタ手段とを含む親テーブルと、それぞれのデュアル・リンクを指すためのデュアル・リンク・ポインタ手段を含むリンク・テーブルと、前記リンクがスパニング・ツリーに現在関与しているかどうかを示すビット位置（STビット）を含む各リンクの参照を割り当てる手段とを含むトポロジ・データベースを記憶する手段と、前記親テーブルとリンク・テーブルの内容を使って、ツリー配列を編成する手段と、各スパニング・ツリーの編成時もしくは再編成時に、前記親テーブルと前記リンク・テーブルとを含むいわゆるトポロジ・データベースを動的に更新する手段とを含む、前記各手段が通常の動作状態の間に動作可能である、データ通信ネットワーク。

（2）各スパニング・ツリーの再編成時に前記親テーブルとリンク・テーブルを更新する前記手段が、（a）当該ノードにおいて、ローカルに記憶された親テーブルと

リンク・テーブルを走査して、子のない第1のノード即ち末端ノードを検出し、前記ノードをルートとして割り当てそれに応じてテーブルを修正する手段と、(b)ローカルの親テーブルとリンク・テーブルとを含むトポロジ・データベースを調査して、ターゲット・ノードをその親とするスパニング・ツリー・リンクを見つけ、当該ターゲット・ノードをもつソース・ノードを親ノードとして選び、それに応じてテーブルを修正する手段と、

(c) すべてのノードが親を獲得しそれに応じてテーブルを修正するまで、ソース・ノードをターゲットとしてその処理を繰り返し、当該ノードのデータベースにおける各ノードの完全な親子関係をツリー構造で定義する手段と、を各ノードに含む上記(1)に記載のデータ通信ネットワーク。

(3) スパニング・ツリーの任意のノードがソース・ノードまたはターゲット・ノードのどちらかとして動作することができ、共にスパニング・ツリーに属するソース・ノードとターゲット・ノードとの間で、要求に応じて迅速な経路決定を行う手段をさらに含み、前記経路決定手段が、ソース・ノードをトリガして、その記憶された親テーブルを上方に走査し、次々に親ノードを上方にリストするソース親リストを生成し記憶する手段と、ソース・ノードをトリガして、その記憶された親テーブルを走査し、ターゲット・ノードから始まって上方にターゲット親リストを生成し記憶する手段と、両方のリストを逆走査し、最後の共通ノードを除いたすべての共通ノードを削除する手段とを含み、残りのソース親リストを順方向に、残りのターゲット親リストを逆方向に単純に連結することによって、前記ソース・ノードと前記ターゲット・ノードの間の経路を決定する、上記(1)または(2)に記載のデータ通信ネットワーク。

(4) スパニング・ツリーのリンク障害時に、元のスパニング・ツリーを、障害リンクの子ノードおよび当該子ノードに接続されているすべてのノードを含む第1のツリーと、元のルートおよび接続されている残りのすべてのノードを含む第2のツリーとに分割して区分し、回復操作を実行して、分割されたツリーを単一のツリーに再び結合し、対応するスパニング・ツリーを再定義する手段をさらに含み、前記回復手段が、

a) 障害リンクの前記子ノードを、前記第1のスパニング・ツリーのルートとして定義する手段と、

b) 元のルートと前記第1のツリーのルートの両方において、現ツリー区画を反映するようにそれぞれのデータベース内のSTビットを再設定する手段と、

c) 前記第1のツリーのルートにおいて、再設定されたトポロジ・データベースを走査して、前記第1のツリーのルートと前記第2のツリーに属するノードとの間のリンクをリストし、予め定義されたリンク特性に基づいて前記リンクのうちの1つを選択して、第1と第2のツリーを1つに接続し、それに応じて前記選択されたリンク

上のターゲット・ノードに通知する手段と、

d) 前記第2のツリーのルート・ノード内で更新済みトポロジ・データベースを走査して、前記ターゲット・ノードと前記第1のツリーのルートとの間のリンクを選択する手段と、

e) ルート移動用の制御データを前記第2のツリーのルートからターゲット・ノードに送り、ツリー合併処理を完了させる手段と、

f) それに応じて各ノードのトポロジ・データベースを更新する手段とを含む、上記(1)ないし(3)のいずれか一項に記載のデータ通信ネットワーク。

(5) 障害リンクの子ノードをI、親ノードをJ、元のスパニング・ツリーのルート・ノードをRとして、ツリーのリンクIJの障害発生後に迅速なスパニング・ツリー回復動作を実施するために、Rをルートとする部分

(R)として定義される区画と、Iをルートとする部分(I)として定義される区画の2つの区画に元のツリーを分割する、上記(1)から(4)のいずれか一項に記載のネットワークにおいて実施される方法であって、ノードIにおいて、Knを部分(R)に属するノードであるとして、トポロジ・データベースを走査してリンクIKnのリストを見つけ記憶するステップと、前記リストにおいて、予め定義されたリンク基準に基づいて、記憶されたリンクKのうち1つのリンクを選択するステップと、前記ノードIにいわゆる結合要求メッセージを前記ノードKに送らせることによって、合併処理を開始するステップと、ノードRにおいて、Knを部分(R)に属するノードであるとして、トポロジ・データベースを走査してリンクKnIのリストを見つけ記憶するステップと、前記予め定義されたリンク基準に基づいて、1組のKnノードの中からリンクKIを選択するステップと、ルート移動メッセージをノードKに送るステップとノードKを使って合併処理を完了し、合併されたツリーのルート機能を獲得し、トポロジ・データベースの導入を再開するステップとを実行するノード管理方法。

(6) 前記合併処理開始ステップが、タイマを始動させて、ノードKに送られる結合要求メッセージの妥当性に制限を設けるステップを含む、上記(5)に記載の方法。

(7) 前記ルート移動メッセージがノードRからノードKに送られる、上記(5)または(6)に記載の方法。

(8) 多数のリンク障害が発生した場合に、リンク障害の発生順に従って操作を同期させる手段を使うことによってスパニング・ツリー回復動作を実施する、上記(5)ないし(7)のいずれか一項に記載の方法。

【0087】

【発明の効果】すでに述べたように、本発明は、各ノードで使用可能な完全トポロジ・データベースを保持しているため、迅速な経路指定あるいはツリーの回復処理を行うことができる。たとえば、リンク障害の場合にも、

所与のスパニング・ツリーの径を小さくし、あるいは所与の低グレードの（または、いわゆる重みの小さい）リンクを、より重みの大きいリンクで置き換えることができる。

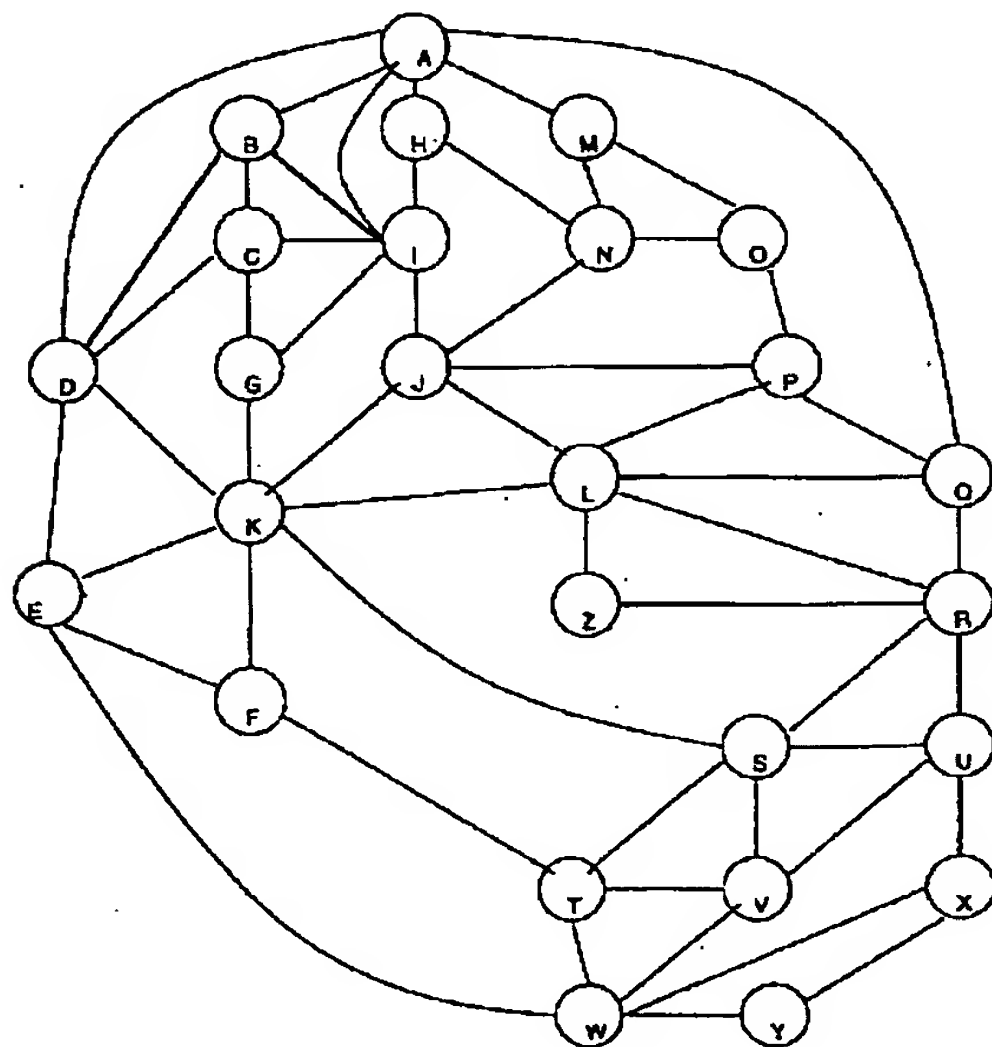
【図面の簡単な説明】

【図1】本発明がその中で実施される従来のデータ・ネットワークの概略図である。

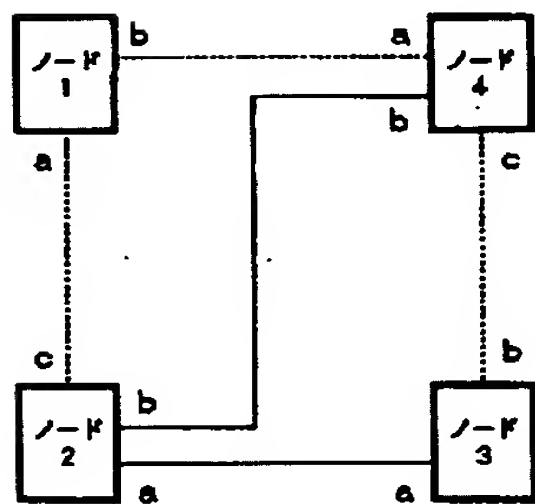
【図2】図1のデータ・ネットワークから誘導され、本発明を実施する手段を備えた、いわゆるスパニング・ツリーの概略図である。

【図3】マルチキャスト動作のネットワーク・ツリーの概略図である。

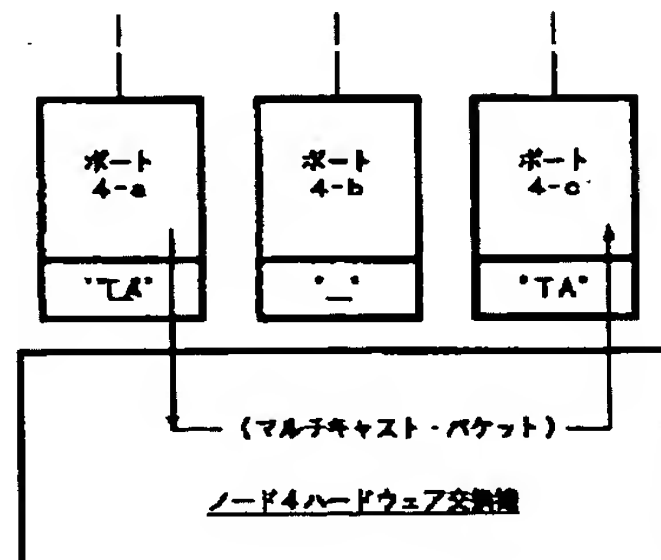
【図1】



【図3】



【図4】



*【図4】図3のネットワーク・ツリーのノードの1つに含まれるハードウェアの概略図である。

【図5】本発明をサポートするために作成されたノード編成の概略図である。

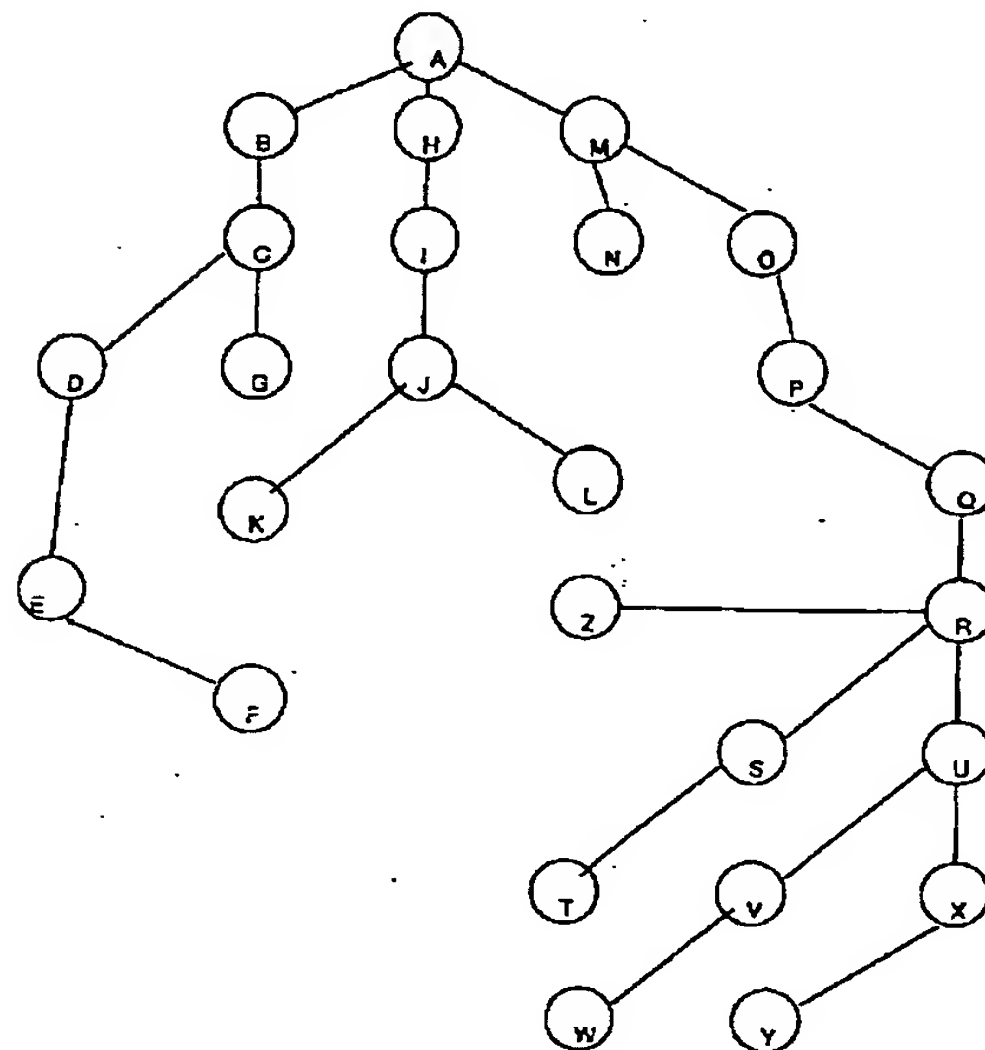
【図6】本発明を実施するために作成されたフローチャートである。

【図7】本発明を実施するために作成されたフローチャートである。

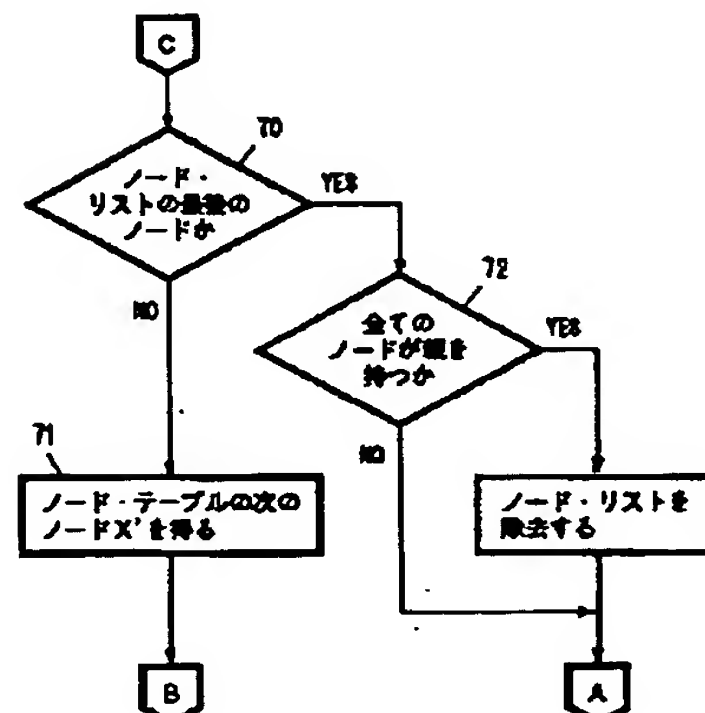
【図8】本発明を実施するために作成されたフローチャートである。

【図9】本発明を実施するために作成されたフローチャートである。

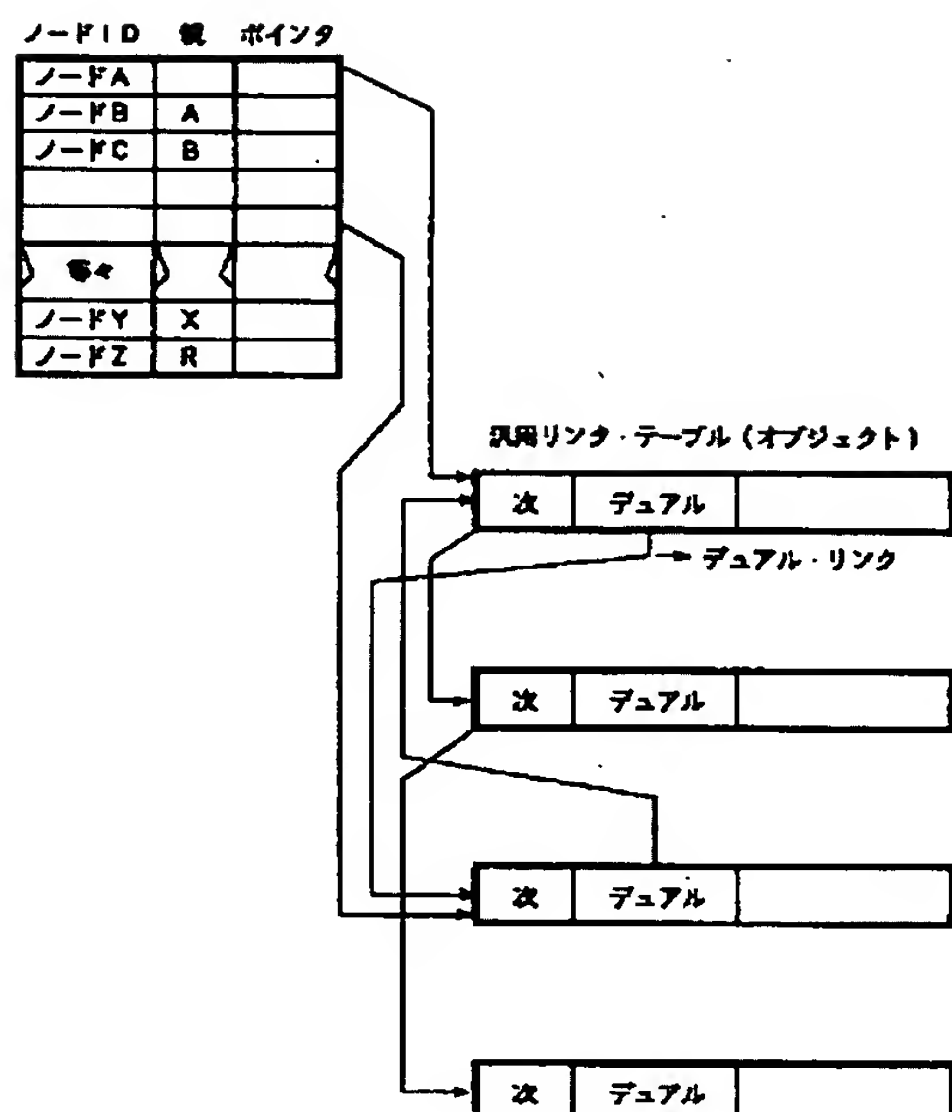
【図2】



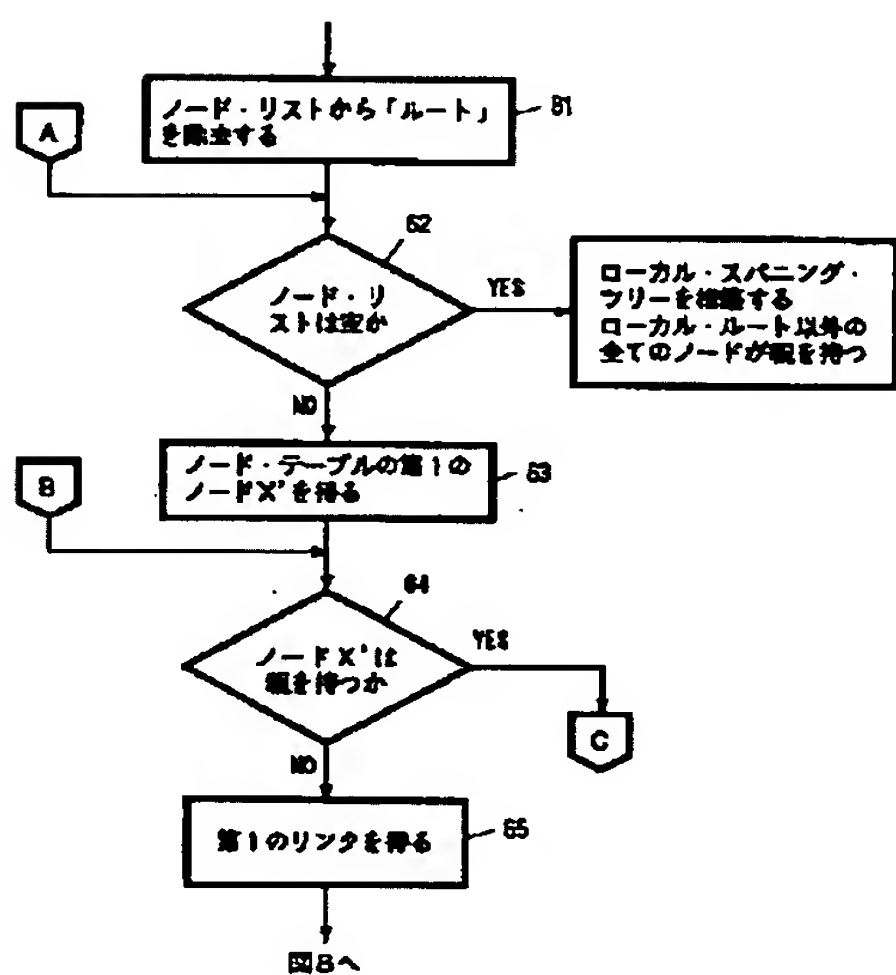
【図9】



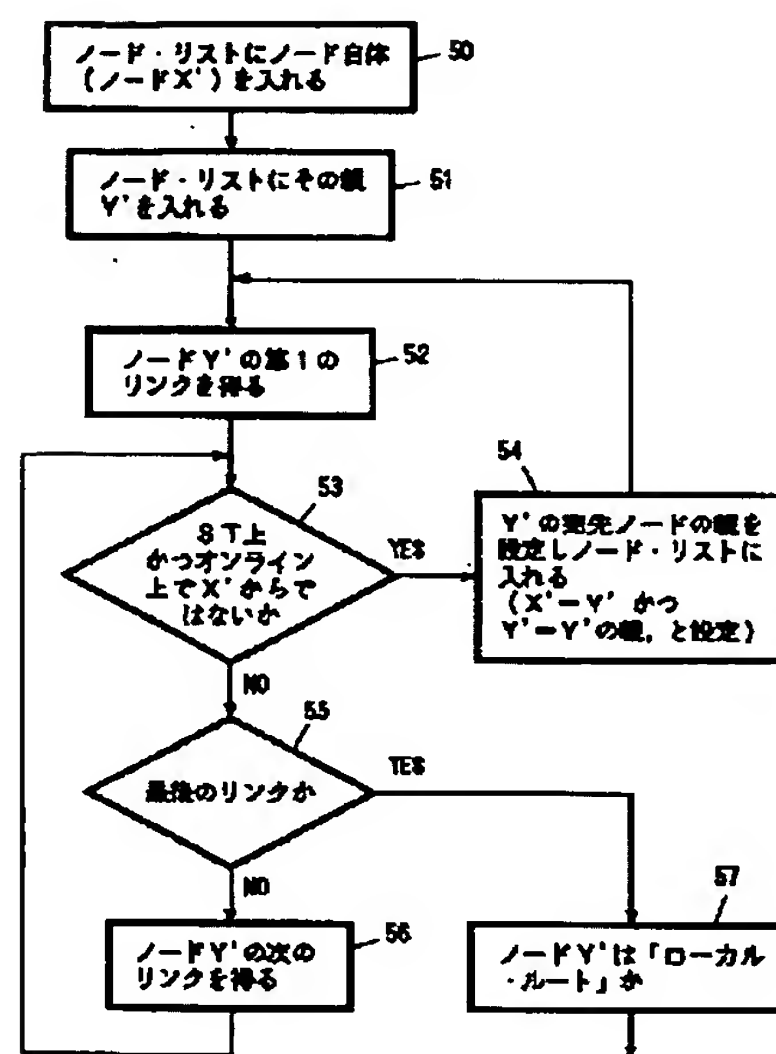
【図5】



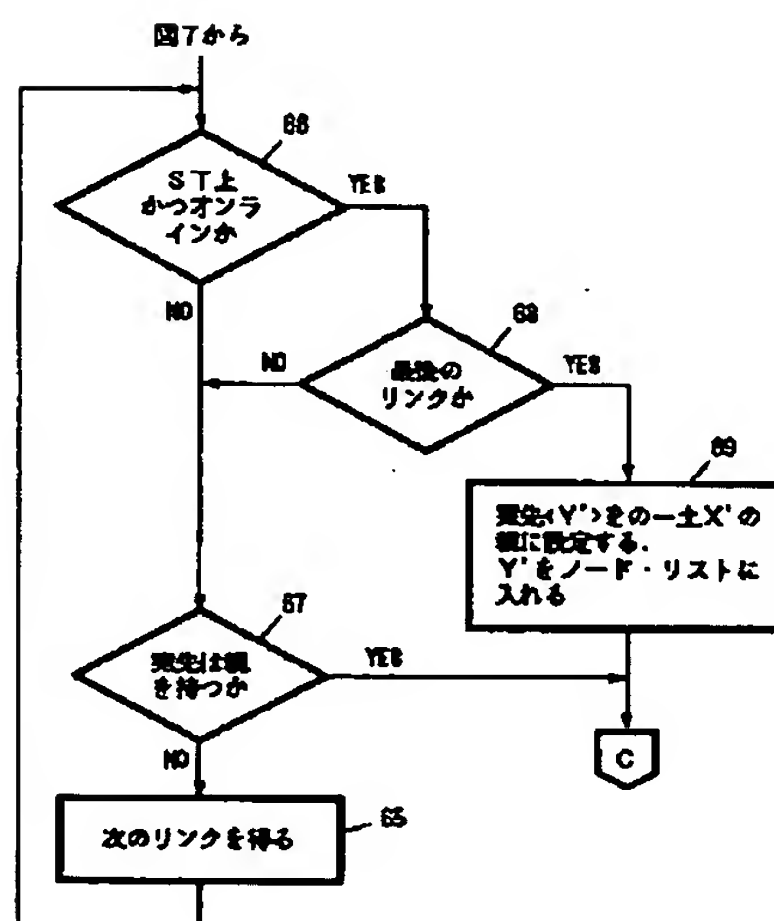
【図7】



【図6】



【図8】



フロントページの続き

(72)発明者 アラン・ブリュヴォ
フランス06220 ヴァローリ アレ・デ・
ミコクリエ 55

(72)発明者 ジャン＝ポール・ショペール
フランス06510 キャロ リュ・ド・ラ・
カニユ 11